

# Network tools and utilities

- tftp on MacOS v26 (Tahoe) [CANCELLED]
- Terminal via serial port on MacOS Sonoma
- Speed tests
  - Global speed tests providers

# tftp on MacOS v26 (Tahoe)

## [CANCELLED]

## preface

tftp was depreciated from macOS.

## lesson (might be useful)

Check is it available

```
ls /usr/libexec/tftpd
```

```
anton-w@ant1mbp7 ~ % ls /usr/libexec/tftpd  
/usr/libexec/tftpd
```

Create local tftp directory AS NORMAL user

```
export u="anton-w"  
export dir="/Users/${u}/delme/tftp"  
mkdir -p ${dir}  
cd ${dir}  
pwd
```

current status

```
netstat -f inet | grep tftp  
ls -li /Users/anton-w/delme/tftp  
ls -nPi | grep ":69"
```

stop service and verify

```
export s="system/com.apple.tftpd"
launchctl stop ${s}
launchctl bootout ${s}
launchctl disable ${s}
launchctl remove ${s}
launchctl list | grep ${s}

netstat -f inet | grep tftp
lsof -iUDP:69
lsof -nPi | grep ":69"
```

default config has disabled option, override is needed

```
export f="/System/Library/LaunchDaemons/tftp.plist"
cat ${f} | grep com.apple.tftpd
cat ${f} | grep -i -A1 disabled
```

Create override

```
export f="/System/Library/LaunchDaemons/tftp.plist"
export fov="/Library/LaunchDaemons/local.tftp.plist"
cp ${f} ${fov}
ls -la ${fov}
vi ${fov}
```

Content

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Disabled</key>
  <false/>
  <key>Label</key>
  <string>local.tftp</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/libexec/tftpd</string>
    <string>-d</string>
```

```
<string>-l</string>
<string>-s</string>
<string>/Users/anton-w/delme/tftp</string>
<string>-u</string>
<string>anton-w</string>
</array>
<key>inetdCompatibility</key>
<dict>
  <key>Wait</key>
  <true/>
</dict>
<key>InitGroups</key>
<true/>
<key>Sockets</key>
<dict>
  <key>Listeners</key>
  <dict>
    <key>SockServiceName</key>
    <string>tftp</string>
    <key>SockType</key>
    <string>dgram</string>
  </dict>
</dict>
</dict>
</plist>
```

## Verify plist file

```
cat ${fov} | grep -i -A1 disabled
plutil -lint ${fov}
```

## Set ownership and permissions

```
ls -la ${fov}
chown root:wheel ${fov}
chmod 644 ${fov}
ls -la ${fov}
```

## Enable service

```
export s="system/local.tftp"
```

```
launchctl stop ${s}
```

```
launchctl bootout ${s}
```

```
launchctl disable ${s}
```

```
launchctl remove ${s}
```

```
launchctl list | grep ${s}
```

```
launchctl enable ${s}
```

```
export f="/Library/LaunchDaemons/local.tftp.plist"
```

```
launchctl load -wF ${f}
```

```
# launchctl bootstrap system ${s}
```

```
launchctl start ${s}
```

```
launchctl list ${s}
```

```
launchctl print ${s} | grep state
```

New file should be in the dump

```
launchctl dumpstate | grep ".plist" | grep ftp
```

```
sh-3.2#  
sh-3.2# launchctl dumpstate | grep ".plist" | grep ftp  
    path = /System/Library/LaunchDaemons/com.apple.ftp-proxy.plist  
    path = /Library/LaunchDaemons/local.tftp.plist  
sh-3.2# █
```

Verify that default TFTP root exists

```
ls -la /private/tftpboot
```

Create if missing

```
sudo mkdir -p /private/tftpboot
```

```
sudo chmod 777 /private/tftpboot
```

Verify tftp is listening on

```
netstat -f inet | grep tftp
```

```
netstat -an | grep "*:69"
```

```
netstat -atp UDP | grep tftp
ls -l /usr/libexec/tftpd
ls -l /usr/libexec/tftpd | grep ":69"
```

## Check state firewall

```
sudo /usr/libexec/ApplicationFirewall/socketfilterfw --getglobalstate
```

if needed, create a rule to allow

```
sudo /usr/libexec/ApplicationFirewall/socketfilterfw --add /usr/libexec/tftpd
sudo /usr/libexec/ApplicationFirewall/socketfilterfw --unblockapp /usr/libexec/tftpd
```

I am using Little Snitch, thus my native firewall is disabled.

TFTP daemon is set up and ready to receive/transmit files.

## troubleshooting

```
launchctl dumpstate | grep ".plist" | grep ftp
```

```
log stream --style syslog --predicate 'process == "launchctl"'
log stream --style syslog --predicate 'subsystem == "com.apple.xpc.launchctl"'
log show --last 5m --style syslog --predicate 'subsystem == "com.apple.xpc.launchctl"'
```

## Start daemon manually

```
❯ /usr/libexec/tftpd \
❯ -d \
❯ -l \
❯ -s \
❯ /Users/anton-w/delme/tftp \
❯ -u \
❯ anton-w
```

File exist, is executable , but does not do anything.

```
pkgutil --file-info /usr/libexec/tftpd
# no package provides
codesign -dv /usr/libexec/tftpd
# has not authority
csrutil authenticated-root status
```

```
file /usr/libexec/tftpd  
mount | grep "sealed"
```

Go home.

# Terminal via serial port on MacOS Sonoma

Connect

Verify connectivity

```
system_profiler SPUSBDataType | grep -i -A5 serial
```

```
anton@ant1mbp4 ~ % system_profiler SPUSBDataType | grep -i -A5 serial
2026-04-01 21:12:57.369 system_profiler[1802:28495] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2026-04-01 21:12:57.370 system_profiler[1802:28495] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2026-04-01 21:12:57.370 system_profiler[1802:28495] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2026-04-01 21:12:57.371 system_profiler[1802:28495] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2026-04-01 21:12:57.371 system_profiler[1802:28495] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
2026-04-01 21:12:57.372 system_profiler[1802:28495] SPUSBDevice: IOCreatePlugInInterfaceForService failed 0xe00002be
    Serial Number: 1[REDACTED]
    Speed: Up to 5 Gb/s
    Manufacturer: Realtek
    Location ID: 0x01334000 / 7
    Current Available (mA): 900
    Current Required (mA): 288
--
    Serial Number: [REDACTED]
    Speed: Up to 12 Mb/s
    Manufacturer: USB-RS232 Interface Converter
    Location ID: 0x01131140 / 14
    Current Available (mA): 500
    Current Required (mA): 50
--
    Serial Number: [REDACTED]
    Speed: Up to 480 Mb/s
    Location ID: 0x01131130 / 13
    Current Available (mA): 500
    Current Required (mA): 500
    Extra Operating Current (mA): 0
```

```
ioreg -p IOUSB
```

```

anton@ant1mbp4 ~ % ioreg -p IOUSB
+o Root <class IORegistryEntry> retain 25>
| +o AppleUSBVHCIBCE Root Hub S <class AppleUSBDevice, id 0x100000554, registered, matched, active, busy 0 (0 ms), retain 14>
| | +o Apple T2 Controller@8016 <class AppleUSBDevice, id 0x10000055b, registered, matched, active, busy 0 (0 ms), retain 13>
| | +o FaceTime HD Camera (Built-in) <class AppleUSBDevice, id 0x10000055c, registered, matched, active, busy 0 (0 ms), retain 13>
| | +o Apple Internal Keyboard <class AppleUSBDevice, id 0x100000569, registered, matched, active, busy 0 (1 ms), retain 19>
| | +o Headset@80400000 <class AppleUSBDevice, id 0x10000056d, registered, matched, active, busy 0 (0 ms), retain 11>
| | +o Touch Bar Backlight@8070 <class AppleUSBDevice, id 0x10000056e, registered, matched, active, busy 0 (0 ms), retain 13>
| | +o Touch Bar Display@806000 <class AppleUSBDevice, id 0x10000056f, registered, matched, active, busy 0 (0 ms), retain 13>
+o AppleUSBXHCI Root Hub Simulator <class AppleUSBDevice, id 0x1000005c9, registered, matched, active, busy 0 (0 ms), retain 10>
| +o USB2.0 Hub <class AppleUSBDevice, id 0x1000005f2, registered, matched, active, busy 0 (0 ms), retain 13>
| | +o 40AV@01120000 <class AppleUSBDevice, id 0x10000062a, registered, matched, active, busy 0 (0 ms), retain 17>
| | +o USB2.0 Hub <class AppleUSBDevice, id 0x10000062b, registered, matched, active, busy 0 (0 ms), retain 14>
| | +o Lenovo Thunderbolt 3 Dock <class AppleUSBDevice, id 0x10000062c, registered, matched, active, busy 0 (0 ms), retain 12>
| | | +o 4-Port USB 2.0 Hub <class AppleUSBDevice, id 0x10000062d, registered, matched, active, busy 0 (0 ms), retain 14>
| | | +o Dell Laser Mouse <class AppleUSBDevice, id 0x10000062e, registered, matched, active, busy 0 (0 ms), retain 15>
| | | +o HD Pro Webcam C920 <class AppleUSBDevice, id 0x10000062f, registered, matched, active, busy 0 (0 ms), retain 19>
| | | +o IOUSBHostDevice <class AppleUSBDevice, id 0x100000630, registered, matched, active, busy 0 (1 ms), retain 11>
| | +o YubiKey OTP+FIDO+CCID <class AppleUSBDevice, id 0x100000631, registered, matched, active, busy 0 (0 ms), retain 15>
+o USB3.1 Hub <class AppleUSBDevice, id 0x100000632, registered, matched, active, busy 0 (0 ms), retain 12>
| +o USB3.0 Hub <class AppleUSBDevice, id 0x100000633, registered, matched, active, busy 0 (0 ms), retain 12>
| +o ThinkPad TBT 3 Dock <class AppleUSBDevice, id 0x100000634, registered, matched, active, busy 0 (0 ms), retain 13>

```

## Identify device

```
ls -ltr /dev/tty*
```

```

crw-rw-rw-  1 root  wheel   0x4000003  1 Apr 20:54 /dev/ttyp3
crw-rw-rw-  1 root  wheel   0x4000002  1 Apr 20:54 /dev/ttyp2
crw-rw-rw-  1 root  wheel   0x4000001  1 Apr 20:54 /dev/ttyp1
crw-rw-rw-  1 root  wheel   0x4000000  1 Apr 20:54 /dev/ttyp0
crw-rw-rw-  1 root  wheel   0x2000000  1 Apr 20:54 /dev/tty
crw-rw-rw-  1 root  wheel   0x9000002  1 Apr 20:54 /dev/tty.Bluetooth-Incoming-Port
crw-rw-rw-  1 root  wheel   0x9000000  1 Apr 20:54 /dev/tty.BLTH
crw--w----  1 anton  tty     0x1000000  1 Apr 21:18 /dev/ttys000
anton@ant1mbp4 ~ %

```

Disconnect, connect back, repeat command to identify it is exactly the same device (it will be usually owned by the root, but by user)

```
ls -ltr /dev/
```

```
crw-r--r-- 1 root wheel 0xa000004 1 Apr 20:54 auditssions
crw----- 1 root wheel 0xb000000 1 Apr 20:54 auditpipe
crw----- 1 root wheel 0x14000001 1 Apr 20:54 afsc_type5
crw-rw-rw- 1 root wheel 0x9000002 1 Apr 20:54 tty.Bluetooth-Incoming-Port
crw-rw-rw- 1 root wheel 0x9000003 1 Apr 20:54 cu.Bluetooth-Incoming-Port
crw-rw-rw- 1 root wheel 0x1e000000 1 Apr 20:54 uart.BLTH
crw-rw-rw- 1 root wheel 0x9000000 1 Apr 20:54 tty.BLTH
crw-rw-rw- 1 root wheel 0x16000000 1 Apr 20:54 nfscInt
crw-rw-rw- 1 root wheel 0x200003d9 1 Apr 20:54 autofs_nowait
crw-rw-rw- 1 root wheel 0x21000011 1 Apr 20:54 autofs_notrigger
crw-rw-rw- 1 root wheel 0x22000011 1 Apr 20:54 autofs_homedirmounter
crw----- 1 root wheel 0x23000000 1 Apr 20:54 autofs_control
crw----- 1 root wheel 0x1f000000 1 Apr 20:54 autofs
crw-rw-rw- 1 root wheel 0x11000000 1 Apr 20:59 random
crw----- 1 root wheel 0x17000001 1 Apr 21:02 bpf1
crw----- 1 root wheel 0x17000000 1 Apr 21:02 bpf0
crw----- 1 anton staff 0 1 Apr 21:12 console
crw-rw-rw- 1 root wheel 0x9000001 1 Apr 21:19 cu.BLTH
crw-rw-rw- 1 root wheel 0x3000002 1 Apr 21:21 null
crw-rw-rw- 1 root tty 0xf000000 1 Apr 21:21 ptmx
crw--w---- 1 anton tty 0x10000003 1 Apr 21:21 ttys003
anton@ant1mbp4 ~ % ls -ltr /dev/
```

Attach `screen` to the serial port

```
screen /dev/ttys003 115200
```

Exit nicely, press

```
CTRL+A
K
Y
```

```
Really kill this window [y/n]
```

# Speed tests

# Global speed tests providers

Remember to open a new test in the new private session window (to ensure new TCP connection will be established)

## CloudFlare

<https://speed.cloudflare.com>

Does measure download for different sizes. Presents jitter in graphical way. Very good

The screenshot displays the Cloudflare Speed Test interface. At the top, it shows the URL `speed.cloudflare.com` and the page title "Speed Test". The main section, "Your Internet Speed", features a large "Download" result of **21.7 Mbps** and an "Upload" result of **420 kbps**. To the right, a "Server Location" map is visible. Below these are "Latency" (44.5 ms), "Jitter" (9.95 ms), and "Packet Loss" (1.50%) metrics. A "Network Quality Score" section indicates "Video Streaming: Average", "Online Gaming: Average", and "Video Chatting: Good". The bottom section, "Download Measurements", includes three graphs for 100kB, 1MB, and 10MB download tests, and "Upload Measurements" for a 100kB test. "Latency Measurements" include "Unloaded latency", "Latency during download", and "Latency during upload". "Packet Loss Measurements" shows a "Packet Loss Test (1000/1000)" with a "Received 99%" bar. The footer contains navigation links (Home, About, Privacy Policy, Terms of Use) and the Cloudflare logo.